Stat260/CS294: Randomized Algorithms for Matrices and Data

Lecture 9 - 10/02/2013

Lecture 9: Fast Random Projections and FJLT, cont.

Lecturer: Michael Mahoney *Scribe: Michael Mahoney*

Warning: these notes are still very rough. They provide more details on what we discussed in class, but there may still be some errors, incomplete/imprecise statements, etc. in them.

9 Fast Random Projections and FJLT, Cont.

We continue with the discussion from last time. There is no new reading, just the same as last class.

Today, we will do the following.

- Show that the two structural conditions required for good LS approximation are satisfied by FJLT projections.
- State the algorithms for fast LS approximation via random projection as well as via random sampling.
- Describe in some more detail the connection between SubspaceJL methods and randomized matrix multiplication.

As a reminder of where we are, to see how we will use these fast random projections and FJLT for RandNLA algorithms, recall from a few classes ago that there were two conditions that were sufficient to obtain relative-error approximation. As a reminder, here are those two conditions.

- **Condition I:**
$$\sigma_{min}(XU_A) \geq 1/\sqrt{2}. \tag{1}$$
- **Condition II:**
$$\left\| U_A X^T X b^{\perp} \right\|_2^2 \leq \frac{\epsilon}{2} \mathcal{Z}^2. \tag{2}$$

Recall also from the last few classes that we can solve LS problems "slowly" with RandNLA methods in one of two ways.

- **Algorithmic Approach I:** with random sampling (which needs some sort of algorithm to compute importance sampling probabilities), or
- **Algorithmic Approach II:** with random projections (which is like uniform sampling if we first preprocess to "flatten out" information in singular value spaces).

In this class and the next, we will show how we can combine these results: in particular, we can establish both conditions with both random sampling and random projection algorithms in $o(nd^2)$ time. Let's start with random projections.

9.1 Establishing the two conditions for fast random projections

Let's start with Condition I, i.e., let's start by establishing a lemma that says that if we uniformly sample in the "randomly rotated" basis then the singular values are all close to 1. (This lemma will be an important ingredient more generally in what follows—basically it means that we are dealing with what is known as a "subspace embedding.")

Lemma 1 *Let $\mathcal{S}$ be a uniform sampling and rescaling matrix. And recall that $\left\|(HDU)_{(i)}\right\|_2^2 \leq \frac{2d\log(40d)}{n}$, for all i. Then if we sample $r \gtrsim O\left(d\log(nd)\log(d\log(nd))\right)$, then with probability ≥ 0.95 we have that*

$$|1-\sigma_i^2\left(\mathcal{S}HDU_A\right)| \leq 1-\frac{1}{\sqrt{\epsilon}}$$

Proof: The idea of the proof is that HD approximately uniformness the leverage scores, so that uniform sampling is approximately optimal, if we are willing to oversample by a factor $1/\beta$, where $\beta \in (0,1]$ quantifies how far from uniform is the leverage score distribution.

In more detail, since $U_A^T DH^T HDU_A = I_d$, we have that

$$\begin{aligned} |1-\sigma_i^2\left(\mathcal{S}HDU_A\right)| &= |\sigma_i\left(U_A^T DH^T HDU_A\right) - \sigma_i\left(U_A^T DH^T \mathcal{S}^T \mathcal{S}HDU_A\right)| \\ &\leq \left\|U_A^T DH^T HDU_A - U_A^T DH^T \mathcal{S}^T \mathcal{S}HDU_A\right\|_2. \end{aligned}$$

Consider the matrix $\left(HDU_A\right)^T$. Since H, D, and U_A are orthogonal matrices, it follows that

$$\begin{aligned} \left\|HDU_A\right\|_2 &= 1 \\ \left\|HDU_A\right\|_F &= \left\|U_A\right\|_F = \sqrt{d}. \end{aligned}$$

Let $\beta = \left(2\log(40nd)\right)^{-1}$, in which case

$$\frac{1}{n} \geq \beta \frac{\left\|\left(HDU_A\right)_{(i)}\right\|_2^2}{\left\|HDU_A\right\|_F^2},$$

$\forall i \in [n]$.

We can now apply the following spectral norm bound theorem:

- If $\left\|A\right\|_2 = 1$, $\left\|A\right\|_F \geq \frac{1}{24}$, $p_i \geq \beta\frac{\left\|A^{(i)}\right\|_2^2}{\left\|A\right\|_F^2}$, and $c \geq \frac{96\left\|A\right\|_F^2}{\beta\epsilon^2}\log\left(\frac{96\left\|A\right\|_F^2}{\beta\epsilon^2\sqrt{\delta}}\right)$, then with probability $\geq 1-\delta$, we have that $\left\|AA^T - CC^T\right\|_2 \leq \epsilon$.

By applying this theorem with $\epsilon = 1-\frac{1}{\sqrt{2}}$ and $\delta = \frac{1}{20}$, then with probability ≥ 0.95, we have that

$$\left\|U_A DH^T HDU_A - U_A DH^T \mathcal{S}\mathcal{S}^T HDU_A\right\|_2 \leq 1-\frac{1}{\sqrt{2}},$$

thus establishing the lemma.

$\diamond$

The above lemma basically says that the first Condition I is satisfied in the randomly rotated space. (Alternatively, it establishes a Subspace JL result.)

Next, let's give a lemma that says that Condition II is satisfied. The following condition will establish the second condition.

Lemma 2 *Assume that* $\left\|(HDU)_{(i)}\right\|_2^2 \leq \frac{2d\log(40nd)}{n}$, *for all* i. *Let* $r \gtrsim 40d\log(40nd)/\epsilon$. *Then, with probability* ≥ 0.9, *we have that*

$$\left\|\left(\mathcal{S}^T HDU_A\right)^T \mathcal{S}^T HDb^{\perp}\right\|_2 \leq \frac{\epsilon \mathcal{Z}^2}{2}$$

Proof: Recall that $b^{\perp} = U_A^{\perp} U_A^{\perp T} b$ and that $\mathcal{Z} = \left\|b^{\perp}\right\|_2$. Since $\left\|U_A^T DH^T HDb^{\perp}\right\|_2^2 = \left\|U_A^T b^{\perp}\right\|_2^2 = 0$, it follows that

$$\left\|\left(\mathcal{S}^T HDU_A\right)^T \mathcal{S}HDb^{\perp}\right\|_2^2 = \left\|U_A^T DH^T \mathcal{S}\mathcal{S}^T HDb^{\perp} - U_A^T DH^T HDb^{\perp}\right\|_2^2$$

We will apply the following approximate matrix multiplication result with probabilities depending on only one matrix:

- If $p_k \geq \beta \frac{\left\|A^{(k)}\right\|_2^2}{\|A\|_F^2}$, then $\mathbf{E}\left[\|AB - CR\|_F^2\right] \leq \frac{1}{\beta c}\|A\|_F^2 \|B\|_F^2$.

Let's let $\beta = (2\log(40nd))^{-1}$, and so for all i we have that

$$\frac{1}{n} \geq \beta \frac{\left\|(HDU_A)_{(i)}\right\|_2^2}{\|HDU_A\|_F^2}.$$

So, we have that

$$\mathbf{E}\left[\left\|(\mathcal{S}HDU_A)^T \mathcal{S}^T HDb^{\perp}\right\|_2^2\right] \leq \frac{1}{\beta r}\|HDU_A\|_F^2 \left\|HDb^{\perp}\right\|_2^2 \leq \frac{d\mathcal{Z}^2}{\beta r},$$

where we have used that $\|HDU_A\|_F^2 = d$. By applying Markov's Inequality, we have that with probability ≥ 0.9 we have that

$$\left\|(\mathcal{S}HDU_A)^T \mathcal{S}^T HDb^{\perp}\right\|_2^2 \leq \frac{10d\mathcal{Z}^2}{\beta r}$$

So, if $r \geq 20\beta^{-1}d/\epsilon$, then with that value of β, the lemma follows.

$\diamond$

So, since these two lemmas were established with the fast Hadamard-based rotations, we now we have all the ingredients for our first "fast" LS approximation algorithm.

9.2 Fast LS approximation

Here, we will describe a random projection algorithm and a random sampling algorithm for approximating the solution to LS that are fast in the sense that they run in $o(nd^2)$ time.

9.2.1 Fast LS approximation via random projections

Here, we will present our first "fast" LS approximation algorithm.

Given as input a matrix $A \in \mathbb{R}^{n \times d}$ a vector $b \in \mathbb{R}^n$, and a number $\epsilon \in (0,1)$, do the following.

1. Let $r = O\left(d\left(\log(d)\right)\left(\log(n)\right) + \frac{d\log(n)}{\epsilon}\right)$. (This holds if $d \leq n \leq e^d$, and we can get messier expressions more generally.)
2. Let S be an $r \times n$ uniform sampling matrix, i.e., it has one nonzero per row, selected u.a.r, with value equal to $\sqrt{n/r}$ and zero otherwise; and choose each of the r rows in i.i.d. trials with replacement.
3. Let $H \in \mathbb{R}^{n \times n}$ be a normalized Hadamard matrix.
4. Let $D \in \mathbb{R}^{n \times n}$ be a diagonal matrix with $\{\pm\}$ entries u.a.r.
5. Compute and return
$$\tilde{x}_{opt} = \left(\mathcal{S}^T HDA\right)^+ \mathcal{S}^T HDb \in \mathbb{R}^d.$$

Here is a theorem that we can establish about this algorithm.

Theorem 1 *The above algorithm gives a vector x_{opt} such that with probability ≥ 0.8 we have that*

- $\|A\tilde{x}_{opt} - b\|_2 \leq (1+\epsilon)\,\mathcal{Z}$
- $\|\tilde{x}_{opt} - x_{opt}\|_2 \leq \sqrt{\epsilon}\kappa(A)\sqrt{\gamma^{-2}-1}\,\|x_{opt}\|_2$

The running time of this algorithm is

$$O\left(nd\log\left(d/\epsilon\right) + d^3\left(\log(d)\right)\left(\log(n)\right) + \frac{d^3\log(n)}{\epsilon}\right)$$

(if $d \leq n \leq e^d$, with a similar but messier expression otherwise).

Proof: Define the following three events:

$$\begin{aligned}
\mathcal{E}_1 &= \text{event that leverage scores are uniformized} \\
\mathcal{E}_2 &= \text{event that singular values approximately equal one} \\
\mathcal{E}_3 &= \text{event that the second matrix multiplication result holds}
\end{aligned}$$

Each of these holds with constant probability, so we can then apply the union bound, which establishes the quality-of-approximation claims.

To do HA takes $O(nd\log(r))$ time, and solving the subproblem takes $O(rd^2)$ time; working though the exact details of those expressions establishes the running time claim.

$\diamond$

More generally, there are several types of "fast" random projection algorithms that take the following form. Given a matrix $A \in \mathbb{R}^{n \times d}$ and a vector $b \in \mathbb{R}^d$,

1. Let Π be any other FJLT.
2. Project A and b onto roughly $r \sim \frac{d\log(d)}{\epsilon}$ rows.
3. Solve $\min_x \|\Pi Ax - \Pi b\|_2$.

In these cases, it can generally be established theorems of the above form that state that you get a $(1 \pm \epsilon)$ approximation in roughly $O(nd\log(r))$ time.

9.2.2 Fast LS approximation via random sampling

Next, let's mention (we'll get into more detail on this next time) a "fast" random sampling algorithm. The basic idea is the following. Given a matrix $A \in \mathbb{R}^{n\times d}$ and a vector $b \in \mathbb{R}^d$,

1. Let $\{p_i\}_{i=1}^n$ be $1 \pm \epsilon$ approximations to the leverage, computed in $o(nd^2)$ time with a black box that we will describe next class.
2. Sample $r \gtrsim \frac{d\log(d)}{\epsilon}$ constraints with probability depending on p_i to construct a nonuniform probability sampling matrix S
3. Solve $\min_x \|SAx - Sb\|_2$.

Again, one can show that you obtain a $(1 \pm \epsilon)$ approximation in roughly $O(nd\log(r))$ time.

- The reason for the running time is that the running time bottleneck for the approximate leverage score computation boils down to a random projection.
- The quality of approximation comes since Condition I and Condition II can be established with this sampling procedure; and the running time is what it is since that is how long it takes to approximate the leverage scores with the black box that we will describe below.

We just mention this now—we'll go into more detail on this next time, as well as discuss why one might prefer sampling versus projection methods.

9.3 More on SubspaceJL and randomized matrix multiplication

Here we will go into a little more detail about the idea of SubspaceJL (Subspace Johnson-Lindenstrauss). This was implicit in what we were doing before, and here we will make it explicit.

First, recall the definition of a JL transform.

Definition 1 *Given $\epsilon > 0$, n points $\{x_i\}_{i=1}^n \in \mathbb{R}^d$, an ϵ-JLT is a $\Pi \in \mathbb{R}^{r\times d}$ such that*

$$(1-\epsilon)\ \|x_i\|_2^2 \leq\ \|\Pi x_i\|_2^2 \leq (1+\epsilon)\ \|x_i\|_2^2$$

As we mentioned before, there are several different constructions for this.

In RandNLA applications, we typically don't want to preserve approximately the distances between a point set of n points, but instead we want to preserve approximately the geometry of an entire subspace. (In addition, we will want it to be "fast," in the sense we use the term before.) That motivates the following definition.

Definition 2 *Given $\epsilon > 0$, an orthogonal matrix $U \in \mathbb{R}^{n\times d}$, where $n \gg d$, $\Pi \in \mathbb{R}^{r\times d}$ is an ϵ-FJLT or a fast subspace JL, if*

- $\left\|I_d - U^T\Pi^T\Pi U\right\|_2 \leq \epsilon$
- $\forall X \in \mathbb{R}^{n\times d}$*, we can compute* ΠX *in* $O(nd\log(r))$ *time.*

From this definition, it is clear that the subspace embedding (as opposed to running time) property of an ϵ-FJLT is a special case (applied to the situation $B = A^T = U$, an orthogonal matrix) of matrix multiplication. (Note that, in this special case, the additive error bounds for matrix multiplication become relative error.) So, randomized matrix multiplication with exact or approximate leverage scores leads to quality-of-approximation bounds of an ϵ-FJLT. (With the approximate leverage score computation algorithm we will discuss next time, we can also satisfy the running time requirements of an ϵ-FJLT.)

Note that, in this case, i.e., when doing sampling, we are looking at the input matrix to construct the importance sampling probabilities. While that satisfies the definition of an ϵ-FJLT, as stated, that is not how TCS people typically think about the problem, since TCS typically thinks about and hopes for data-oblivious embeddings/projections. (In other settings, there are other algorithmic advantages in doing that.) Importantly, since random projections uniformize leverage scores to permit uniform sampling in the randomly rotated basis, we can achieve this. This holds for both slow and fast random projections, with appropriate parameter setting; and this may be viewed as providing data-oblivious approximate matrix multiplication (via projections rather than sampling). Here is a lemma we can show about this with FJLTs.

Lemma 3 *Let* $\hat{H}_1 = 1$, $\hat{H}_{2n} = \begin{pmatrix} \hat{H}_n & \hat{H}_n \\ \hat{H}_n & -\hat{H}_n \end{pmatrix}$, *and* $H_n = \hat{H}/\sqrt{n}$. *Let* $\Pi = S^THD$, *where* S^T *is a uniform sampling and rescaling operator that chooses r rows from HD, and let $U \in \mathbb{R}^{n\times d}$ be a (fixed but arbitrary) orthogonal matrix. Then, if* $r \geq O\left(\frac{d\log(nd)}{\epsilon^2}\log\left(\frac{d\log(nd)}{\epsilon^2}\right)\right)$, *then with probability* ≥ 0.9 *it follows that* Π *is an ϵ-FJLT for U.*

Proof: Combine the two results:

- The previous lemma that says: $\max_{i\in[n]} \left\|(HDU)_{(i)}\right\|_2^2 \leq \frac{2d\log(40nd)}{n}$.
- The previous lemma that says: $|1 - \sigma_i^2\left(S^THDU\right)| \leq 1 - \frac{1}{\sqrt{2}}$.

The lemma follows.

◇

There are many different constructions that, for appropriate parameter settings, satisfy the Subspace JL property, including the following.

- "fast" Hadamard-based constructions
- "slow" Gaussians, $\{\pm 1\}$ r.v.s, etc.
- Random sampling matrices with, e.g., $p_i \sim \frac{1}{d}\left\|U_{(i)}\right\|_2^2$. This can be computed exactly which is "slow," or this can be computed approximately with an algorithm we will discuss next time which is "fast."

9.4 An aside for next class

Next time, we will go into more detail on the random sampling algorithm. In particular, we are going to prove that we can compute the leverage scores quickly, if we are willing to settle for approximations. To do so, we will use the following lemma (although we could prove it in other related ways). This lemma states several related results that are related to having a good subspace embedding and that are needed to get good bounds for LS-related problems, and it highlights a key property of the Moore-Penrose generalized inverse that is false in general but that is true when the subspace is preserved.

Lemma 4 *Let $A \in \mathbb{R}^{n\times d}$, with $n \gg d$, and let rank$(A) = d$, nd let $A = U\Sigma V^T$ be the SVD of A, and let S satisfy the Subspace JL property. (E.g., it could be a sampling matrix constructed with leverage-based sampling probabilities, or it could be a data-agnostic random projection matrix, that is either fast or slow.) Then,*

- *rank(SA) = rank(SU) = rank(U) = rank(A) = d*
- $\left\|\Sigma_{SU} - \Sigma_{SU}^{-1}\right\|_2 \leq \epsilon$
- $(SA)^+ = V\Sigma\,(SU)^+$
- $\left\|(SU)^+ - (SU)^T\right\|_2 = \left\|\Sigma_{SU} - \Sigma_{SU}^{-1}\right\|_2$

Proof: For the first claim, note that $\forall i \in [\rho]$ we have that

$$\begin{aligned}\left|1 - \sigma_i^2(SU)\right| &= \left|\sigma_i(U^TU) - \sigma_i(U^TS^TSU)\right| \\ &\leq \left\|U^TU - U^TS^TSU\right\|_2 \\ &\leq \epsilon,\end{aligned}$$

with appropriate parameters if $r \gtrsim \frac{d\log(d)}{\epsilon}$. For the second claim, note that

$$\begin{aligned}\left\|\Sigma_{SU}^{-1} - \Sigma_{SU}\right\|_2 &= \max_{ij\in[\rho]}\left|\sigma_i(SU) - \frac{1}{\sigma_j(SU)}\right| \\ &= \max_{ij\in[\rho]}\frac{|\sigma_i(SU)\sigma_j(SU) - 1|}{|\sigma_j(SU)|} \\ &\leq \max_{j\in[\rho]}\frac{\left|\sigma_j^2(SU) - 1\right|}{\sigma_j(SU)} \\ &\leq \frac{\left\|U^TU - U^TS^TSU\right\|_2}{\sqrt{1 - \left\|U^TU - U^TS^TSU\right\|_2}},\end{aligned}$$

where the last inequality follows since

$$\frac{1}{\sigma_i(SU)} \leq \frac{1}{\sqrt{1 - \left\|U^TU - U^TS^TSU\right\|_2}},$$

which follows since $|1 - \sigma_i(SU)| \leq \left\|U^T U - U^T S^T S U\right\|_2$. For the third claim, note that

$$\begin{aligned}
(SA)^\dagger &= \left(SU_A \Sigma_A V_A^T\right)^\dagger \\
&= \left(U_{SA} \Sigma_{SA} V_{SA}^T \Sigma_A V_A^T\right)^\dagger \\
&= V_A \left(\Sigma_{SU} V_{SU}^T \Sigma_A\right) U_{SA}^T \qquad (3) \\
&= V_A \Sigma_A^{-1} V_{SU} \Sigma_{SU}^{-1} U_{SU}^T \qquad (4) \\
&= V_A \Sigma_A^{-1} (SU)^\dagger .
\end{aligned}$$

(Note that $(SA)^\dagger = V_A \Sigma_A^{-1} (SU)^\dagger$ might seem intuitive, given the behavior of inverses, but it is false in general, and it it a common mistake to assume that it it true of generalized inverses. In particular, note that Eqn (4) holds since rank is preserved; otherwise it is false, and the claims of the lemma fail to hold. On the other hand, Eqn (3) holds for any orthogonal matrices.) For the fourth claim, note that

$$\begin{aligned}
\left\|(SU)^\dagger - (SU)^T\right\|_2 &= \left\|\left(U_{SU} \Sigma_{SU} V_{SU}^T\right)^\dagger - \left(U_{SU} \Sigma_{SU} V_{SU}^T\right)^T\right\|_2 \\
&= \left\|V_{SU} \left(\Sigma_{SU}^{-1} - \Sigma_{SU}\right) U_{SU}^T\right\|_2 \\
&= \left\|\Sigma_{SU}^{-1} - \Sigma_{SU}\right\|_2 ,
\end{aligned}$$

where the last claim follows since V_{SU} and U_{SU} have orthogonal columns. The lemma then follows. $\diamond$

So, this lemma is just another way to prove the same LS result. (In fact, this was the way that we first established the relative-error LS result.) Basically, it establishes several senses in which $(SU)^\dagger \approx (SU)^T$ if rank is preserved.

This lemma was stated for completeness, since we could prove the leverage score result directly. Alternatively, we could prove the previous theorem with this result. Part of the reason for doing it this way is that I didn't have a chance to get a uniform set of notes before class, but part of the reason for this is also since using this following lemma is perhaps more natural for NLA people as opposed to TCS people, since it highlights the role of the singular structure of the input matrix.